



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/824,251	04/13/2004	Andreas W. Wendker	APLE P0004C	6431
62234 7590 04/27/2009 ADELI & TOLLEN, LLP 11940 San Vicente Blvd., Suite 100 LOS ANGELES, CA 90049				
EXAMINER MYINT, DENNIS Y				
ART UNIT 2162		PAPER NUMBER		
MAIL DATE 04/27/2009		DELIVERY MODE PAPER		

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary

Application No.

10/824,251

Applicant(s)

WENDKER ET AL.

Examiner

DENNIS MYINT

Art Unit

2162

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 01/06/2009.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 20-27, 31, 32, 35-38, 42-44, 46-53, 56-63 and 75-88 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☒ Claim(s) 42-44 and 84-86 is/are allowed.
- 6) ☒ Claim(s) 20-27, 31-32, 35-38, 46-53 and 56-63, 75-83, and 87-88 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. _____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- 1) ☒ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-848)
- 3) ☐ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____
- 4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date _____
- 5) ☐ Notice of Informal Patent Application
- 6) ☐ Other: _____

DETAILED ACTION

1. This communication is responsive to Applicant's Amendment, filed on January 16, 2009.
2. Claims 20-27, 31-32, 35-38, 42-44, 46-53 and 56-63, and 75-88 are currently pending in this application. Claims 20, 31, 35, 42, 46, 52, 58, 59, 80, 84, and 87 are independent claims. In the Amendment filed on January 16, 2009, claims 20-22, 31, 35, 37, 42, 44, 46, 49-52, 58, 59, 80, 81, 83, 84, 86, and 87 were amended. **This office action is made final.**
3. In light of the amendments made to claims 31, 42, 58, 59, and 80 and Applicant's explanations (Applicant's arguments, pages 18-19), objection to the specification, objection to the drawing(s) and rejection of claims 31-32, 77-78, 42-43, 46-51, 52-53, 56-57, 58, 63, 59, 60, 62, and 80-82 under 35 U.S.C. § 101 in the prior office action are hereby withdrawn.
4. Examiner interprets the "computer-readable medium" as recited in claims 31 and 42 to be "memory" as recited in lines 1-5 of page 30 of the specification (i.e., "*As either client or server, computer system 510 has a processor 512 and a memory 511, such as RAM, which is used to store/load instructions, addresses and result data as desired*"). Examiner interprets "the system" as recited in claim 46 to be "the computer system" as depicted in Figure 5 of the specification and described in 1-5 of page 30 of the specification.

Response to Arguments

5. Applicant's arguments filed on January 16, 2009, have been considered but are moot in view of new ground(s) of rejection.

Claim Rejections - 35 USC § 103

6. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

7. This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).
8. Claim 20, 24-26, 31, 35-37, 46-50, 52-53, 56-58, 59-62, 63, 75-83, and 87-88 are rejected under 35 U.S.C. 103(a) as being unpatentable over Galea (hereinafter "Galea", U.S. Patent Number 6404445) in view of Everett (U.S. Patent Application Publication

Number 2004/0024790) and further in view of Guyon et al., (hereinafter "Guyon", U.S. Patent Application Publication Number 2006/0064415) and further in view Cason (U.S. Patent Number 6035300).

As per claim 20, Galea is directed to a method for creating a description of a user interface that transacts with a database having a data model comprising a plurality of entities, (Galea, Figure 3 and Column7 Lines 23-41, where it is clear that a data model or GUI database 302 is received by the modeler 304) and teaches the limitations:

a) **"receiving a request from the user interface from a client"** (Galea, Column 5, Lines 51-53, i.e., *a user accesses a server through client 106, 108 via wide area network 112 in order to access items for sale (domains) in the product and GUI database*);

b) **"after receiving said request identifying (classifying) the plurality of entities and entity types"** (Galea, Column 5 Lines 23-50, i.e., *In one embodiment, the databases contain compiled product or server components (domains). Each domain contains descriptive information which bind the domain elements to their respective GUI representations. Prerequisite tags may indicate the **GUI element type** such as, for example, single select **list box**, **multi-select list box**, **radio buttons**, **click boxes**, and **input text field**, or images that present the selections options for that domain. Optional domain tags determine the run time behavior of each GUI element such as, for example, whether. or not to display invalidated options (and subsequently allow single selections overrides), whether or not user interface pages are presented as sequential pages or as grouped elements, whether or not pages/groups for each domain are to be*

displayed and where they are to be displayed, whether or not the domain is hidden (not included in the user interface), whether or not the domain is displayed as a read only form element (whereby the domain selection may only be made through configuration logic), whether or not to include local or specific domain and domain options representations, whether or not and which corresponding images are to be displayed with each option selection, whether or not and in which form an image element is to be updated with the corresponding image, whether or not and for which corresponding help pages are available for each domain and option selection, whether or not and in what form a status text to offer when navigation to each domain is included, and whether or not the domain is required for final product selection (order inclusion; This disclosure of Galea clearly teaches that data model contained in the GUI database 302 contains a plurality of entity types, that is, the database contains compiled product or server components (also referred to as domains and each of these domains contains descriptive binding domain elements to their respective GUI representations. Therefore, domains are entities and each entity describes a type of data structure such as various graphical user elements such as radio buttons, click boxes, input text fields etc.);, the classifying comprising:

*"(iteratively determining whether each entity in the plurality of entities satisfies a set of conditions; and when an entity satisfies the set of conditions, classifying the entity as a particular) **"entity type for which a user interface element is created within a particular window of a set of windows, wherein the user interface element provides access to said entity through the user***

interface" (Galea, Figure 9; Note that user interface elements are created within a set of windows and users can access to the entities contained in the GUI database 302 via said user interface elements such as list box, multi-select list box, radio buttons, click boxes, and input text field))

c) **"creating the description of the user interface (based upon the classification of the plurality of entities"** (Galea, Column 5 Lines 51-59, i.e., *A user accesses a server through client 106, 108 via wide area network 112 in order to access items for sale (domains) in the product and GUI database. Each product or service configuration domain, together with the user interface tags, is compiled into a secure, binary compressed file format. After a user accesses the domain through client 106, 108, the compiled configuration model is downloaded to the browser of client 106, 108. The user interface is dynamically generated based upon the compiled domain tags*).

Galea teaches "a plurality of entities" based upon which user interfaces are created. However, Galea does not explicitly teach "classifying the plurality of entities into entity types, the classifying comprising: iteratively determining whether each entity in the plurality of entities satisfies a set of conditions; and when an entity satisfies the set of conditions, classifying the entity as a particular entity type; and d) distributing the description to said client in order to allow said client generate user interface elements of said user interface that permits a user to interact with said database.

On the other hand, Everett teaches the limitations:

"classifying the plurality of entities into entity types, the classifying

comprising: (iteratively) **determining whether each entity in the plurality of entities satisfies a set of conditions; and when an entity satisfies the set of conditions, classifying the entity as a particular entity type;**" (Everett, **Paragraph 0090**, i.e., "In its preferred embodiment, the KnOS Vector Key has four dimensions of location embodied therein. They are called Environment, Repository, Context and Item, and will be expressed herein as [E,R,C,I]. If "White Beans" (a Data Instance) were inserted into the "Stock" Context at the "Manhattan South" Repository of the "Global" Environmental domain, "White Beans" might be registered in the KnOS with the Vector Key [E,R,C,I]=[0,6,1,3]"; Note that said vector keys [E,R,C,I] of Everett maps to "the set of conditions" of the claimed invention based upon which entity types are classified; Everett, **Paragraph 0092**, i.e., "Structural Symmetry--The target Vector Key of each bi-directional pair of binary associations is stored fully encapsulated with its subject Data Instance. If "White Beans" (subject) has-price-of (relationship) "\$13.45" (target), the Item holding the target dollar amount might get registered in the KnOS as [E,R,C,I]=[0,6,2,3]. The subject Item [0,6,1,3] ("White Beans") would be encapsulated with the target Item [0,6,2,3] (for \$13.45), and vice versa. These type -classified sets of encapsulated references for each Item are stored in ordered one-dimensional arrays, called Vector Key Sets (VK Sets) by context type, as shown in Drawing 25, Structural Symmetry--KnOS Items. Every Item has the same fully symmetrical database structure--a) a self-reference (Vector Key), b)a Data Instance, c) Relationships (expressed as VK Sets), and d) (optionally) one or more Item Embedded Elements (IEE). KnOS Item structures, with all of their encapsulated target References, **are also classified by type** and stored

in classified Contexts, for example, 'Stock' in one Context and 'Price' in another, as shown in Drawing 26, Structural Symmetry--KnOS Contexts";).

At the time the invention was made, it would have been obvious to a person of ordinary skill in the art to modify the method of Galea to add the feature of classifying entity types, as taught by Everett, to the method of Galea so that the resultant method would teach "classifying entity types" based on a set of conditions (vector keys). One would have been motivated to do so in order to improve data operations on databases (Everett, Paragraph 0091, i.e., *"they can be classified to improve efficiency"*).

Galea in view of Everett does not explicitly teach the limitation:

"iteratively determining", and "d) distributing the description to said client in order to allow said client generate user interface elements of said user interface that permits a user to interact with said database".

On the other hand, Guyon teaches the limitation:

"iteratively determining (Guyon, paragraph 0137, i.e., **"Recursive Feature Elimination (RFE) methods can be used to overcome this problem. RFE methods comprise iteratively 1) training the classifier, 2) computing the ranking criterion for all features, and 3) removing the feature having the smallest ranking criterion ; Also note Figure 3 of Guyon wherein classifiers are "iteratively determined")**

At the time the invention was made, it would have been obvious to a person of ordinary skill in the art to modify the method of Galea in view of Everett to add the feature of iteratively determining data items, as taught by Guyon, to the method of Galea in view of Everett , so that the resultant method would iteratively determine entity

types whether they meet a set of conditions (vector keys). One would have been motivated to do so in order to remove data items with do not meet certain criteria (Guyon, paragraph 0137, i.e., *"removing the feature having the smallest ranking criterion"*).

Galea in view of Everett and further in view of Guyon does not explicitly teach the limitation: "d) distributing the description to said client in order to allow said client generate user interface elements of said user interface that permits a user to interact with said database".

On the other hand, Cason teaches said limitation, among others, as follows:

a) **"receiving a request for the user interface from a user"** (Cason, Figure 1 and Column 3 Lines 29-31, i.e., *Referring further to Fig. 1, the specific panel displays 122 and 124 provide examples of **graphical user interface(GUI) representations of the data model entities** 102 and 104, respectively; Cason, Figure 10A i.e., **INITIALIZATIONS & DEFAULTS** and Column 9 Lines 22-26, i.e., *Referring now to FIG. 10, a block diagram illustration of a preferred embodiment of the invention is described from the view of the user at terminal 200. In this embodiment, **initialization and defaults** block 202 references online help file 204, screen layouts & bitmaps 206, user custom table 217, and start from selection list 210; Note that by initializing the system/method, the user herein is requesting to obtain a graphical interface representation of database models 122 and 124);**

b) **"after receiving said request, identifying the plurality of entities into entity types"** (Cason Column 4 Lines 29-57, i.e., *In an ERwin diagram, the entity is*

represented by drawing a box 280 with the name of the entity on top and all of the attributes of the entity inside the box. The horizontal line 281 in the entity box divides the attributes into two sets: keys and non-keys (or, data). The area above the line is the key area 282, and the area below the line is the data area 284. **The key is the set of attributes that identifies an entity.** A key attribute is an attribute which, either by itself, or in combination with other key attributes, forms a unique identifier for an entity. Primary keys are placed above the line in the key area. A non-key attribute is an attribute which has not been chosen as a key, and is placed in the data area. **Entity instances in an ERwin model are always identified by key attributes.** Key attributes are **chosen** from a set of candidate key attributes, which are attributes, or groups of attributes, which can be selected as **primary keys**, and as a candidate key must uniquely identify each instance of the entity, and may have no part null, i.e., empty or missing. In choosing a primary key from the set of candidate key attributes, **the data modeler tries to find an attribute which will not change its value over the life of each instance of the entity**, is reasonably small, and is not an intelligent key. An intelligent key is a key where the structure of the digits in the identifier indicates groupings, locations, classifications, dates or the like. Candidate keys not selected as primary keys are designated as alternate keys, and the symbol AK_n, where n is a number, is placed after those attributes which form the **alternate key**. Alternate keys are used to show different indexes the business will use to access the data; Herein, **key attributes** of database entities "map" to the "entities" of the claimed invention, which are classified as primary keys, alternate keys and so on); and **"the classifying**

comprising:

"determining whether a first entity in the plurality of entities satisfies a set of conditions" (Cason, Column 4 Lines 29-57, i.e., *Key attributes are **chosen** from a set of candidate key attributes, which are attributes, or groups of attributes, which can be selected as **primary keys**, and as a candidate key must uniquely identify each instance of the entity, and may have no part null, i.e., empty or missing; Herein, "a primary key" is a first entity which satisfies a set of attributes. Note that a primary could be a combination of attributes); **and***

"classifying the first entity as a particular entity type upon determining that the first entity satisfies the set of conditions" (Cason, Column 4 Lines 29-57, Primary Keys are a first entity type and alternate keys and other types of keys are a second or other type of keys);

c) "creating the description of the user interface based upon the classification of the plurality of entities" (Cason, Column 11 Lines 10-31, i.e., *In accordance with the invention, **each user's database 218** will include one or more **description entities** (see, for example, **table definition 304** and **field definition 306**, FIG. 11) and each relationship is given a type from a set including **the type "description"**, as will be more fully described in connection with FIGS. 11-16. **These entities include a description of a particular record** containing information in a plurality of leafs, one leaf for each record, necessary to enforce the cardinality rules. In this embodiment, invoke a relationship control 236 checks the type of the relationship linking the entity selected to the entity displayed and, as shown at test 240, invokes*

page editor 238 if the relationship has type "text description". **Page input editor 238 opens at terminal 219, allowing the user to enter to the description entity instructions** which define the cardinality rules. Page input editor 238 uniquely numbers the lines of the leaf being edited thus providing keys to the lines of data in the description, closes out the window and returns control to record edit interface 220. Alternatively, the user may activate cancel 250 key 250 to return control to invoke relationships 236 without saving any changes made to the selected leaf of the description entity being edited; **Carson** Column 12 Lines 5-25, i.e., Referring now to FIGS. 11-16, the ERwin diagrams describing the preferred embodiment of the invention will be described. These FIGS. 11-16 include entity-attribute-relationship (EAR) models of FIGS. 11, 13 and 15 and physical models of FIGS. 12, 14 and 16. FIGS. 11 and 12 describe the database model. FIGS. 13 and 14 describe **the user interface model**. FIGS. 15 and 16 combine the database and user interface models. The EAR model contains the database model which describes the invention. A physical model describes an actual implementation of an EAR model of the invention for a given database, and includes table names, field names within tables, and their type and size. The relationships in a physical model are the same as in the EAR model. **The user interface model describes the look and feel, which includes the "type of control" to be invoked, the type of window to be displayed, and the default values to be placed in the fields as the user defines a database model for some aspect of a business or other organization or process** (such as the database example set forth in connection with FIG. 7) and thereafter works with that database to add, delete,

change records or modify the database structure itself; Note herein that "the user interface model" of Cason maps to the "description of the user interface" of the claimed invention. Said "user interface model" of Carson is derived based on "keys" which uniquely identifies database entities from a single record to a table. As such, said "user interface model" includes "descriptions" (Carson Column 11 Lines 10-31) and are derived based on keys); and

"distributing the description to said client in order to allow said client generate user interface elements of said user interface that permits a user to interact with said database" (Cason, Column 10 Lines 51-67, i.e., *Change record key 226 processes a request by the user to edit and change a primary key value. This is a touchy point in data bases, **because everything is driven by the primary key(s)**. Usually, controls are in place to prevent changing such primary keys because there are many pointers to that key. In accordance with this preferred embodiment of the invention, change record key 226 initiates a global change, and all pointers to the record identified by the primary key being changed are accessed and updated. For example, if a primary key is a name, and person identified by that name marries, then the key would need to be changed. In those instances where records are stored in user's database 218 in alphabetical, or some other such order, get next record key 230 and get previous record key 232 are provided for accessing the next and previous records, respectively; Cason Column 11 Lines 1-9, i.e., In response to the identification of the entity selected, meta data file 216 is accessed to determine which relationships exist for it. There will be one control established at terminal 219 for each relationship defined in*

meta data file 216 for the selected entity. Just as each entity has one or more primary keys, relationships have keys that link entities. A control may be, for example, a push button, popup menu, icon, function key, number/enter key-- whatever it takes to start an action; Carson Column 12 Lines 5-25, i.e., The user interface model describes the look and feel, which includes the type of control to be invoked, the type of window to be displayed, and the default values to be placed in the fields as the user defines a database model for some aspect of a business or other organization or process (such as the database example set forth in connection with FIG. 7) and thereafter works with that database to add, delete, change records or modify the database structure itself; Note that any changes to database entities and their respective keys as well as relationship(s) between database entities (including respective "types of controls") are stored in the "user Interface mode" of Cason, from which a graphical user interface is generated which allows the user to add, delete, change records or modify the database structure). Note Figure 8 of Cason wherein the user terminal is separated from the database. Also note Column 8 Lines 19-40, wherein "presentation manager" and "presentation manager code" on the client generates "graphical user interface" which represent the database.

At the time the invention was made, it would have been obvious to a person of ordinary skill in the art to add the feature of classifying database entities into a first entity type if said entities meets a set of conditions, as taught by Cason, to the method of Galea so that, in the resultant method, database entities would be classified into a first entity type if said entities meets a set of condition and a description of user interface

would be created based on said classification. One would have been motivated to do so in order to provide a "modeling system that takes entities, attributes, and relationships (with cardinality and optionality on data models) and generates a working user interface" (Cason, Column 8 lines 15-19).

As per claim 24, Official Note is taken that writing descriptions in XML is notoriously well known in the art. Galea also teaches XML format in Column 8 Lines 25-67 and Column 9 Lines 1-22.

As per claim 25, Galea in view of Everett and further in view of Guyon and further in view of Cason teaches the limitation:

"wherein the classifying and creating are performed automatically without human assistance" (Galea, Column 6 Lines 20-34, i.e., *The distributed client-side e-commerce service system 100 allows for the **automatic generation** of a user interface for a particular product and/or service model **"on-the-fly"** such that a web user is always presented with the latest product or service selection choices*).

As per claim 26, Galea in view of Everett and further in view of Guyon and further in view of Cason teaches the limitation:

"before the classifying, obtaining a current data model of the database, the current data model reflecting any changes to the database up to when the current data model is obtained, wherein a current description of the user interface is

created using the current data model of the database" (Galea, Column 6 Lines 20-34, i.e., *The distributed client-side e-commerce service system 100 allows for the automatic generation of a user interface for a particular product and/or service model "on-the-fly" such that a web user is always presented with the latest product or service selection choices*). In Column 10 Lines 44-57, Galea teaches database modeling. From these teachings, it can be inferred that Galea obtains a current data model of the database.

As per claim 27, Galea in view of Everett and further in view of Guyon and further in view of Cason teaches the limitation:

"after sending the description, retrieving the data from the database, said data for populating at least one generated user interface element ; and sending the data to the client, said data sent prior to receiving any transaction from said user" (Galea, Figure 7 and Column 6 Lines 20-34). Galea teaches the client request as show in Figure 7 as well as described in Column 6 Lines 20-34, in an e-commerce system that *"allows for the automatic generation of a user interface for a particular product and/or service model "on-the-fly" such that a web user is always presented with the latest product or service selection choices."* It can be inferred that the use and presentation of the product and service selections of the e-commerce taught by Galea can be interpreted as a request for creation of a description of a user interface onto a client system.

Claim 31 is essentially the same as claim 20 except that it set forth the claimed invention as a computer readable medium rather than a method and rejected for the same reasons as applied hereinabove.

As per claim 35, Galea in view of Everett and further in view of Guyon and further in view of Cason is directed to a method for generating a user interface on a client computer system that transacts with a database of a server computer system, said database having a data model containing a plurality of entities (Galea, Figure 3 and Column7 Lines 23-41, where it is clear that a data model or GUI database 302 is received by the modeler 304) and teaches the limitations:

"sending a request from the user interface from the client computer system to the server computer system" (Galea, Figure 3 and Column7 Lines 23-41; Galea, Column 5 Lines 51-59; Galea, Column 5 Lines 23-50:);

"receiving a description of the user interface at the client computer" (Galea, Column 5 Lines 51-59, i.e., *A user accesses a server through client 106, 108 via wide area network 112 in order to access items for sale (domains) in the product and GUI database. Each product or service configuration domain, together with the user interface tags, is compiled into a secure, binary compressed file format*) **"the description based upon a set of conditions that classify the plurality of entities into entity types, said classification initiated by the request"** (Galea in view of Everett as applied to claim 20 above; Galea, Column 5 Lines 23-50, i.e., *In one embodiment, the databases contain compiled product or server components (domains).*

*Each domain contains descriptive information which bind the domain elements to their respective GUI representations. Prerequisite tags may indicate the **GUI element type** such as, for example, single select **list box**, **multi-select list box**, **radio buttons**, **click boxes**, and **input text field**, or images that present the selections options for that domain. Optional domain tags determine the run time behavior of each GUI element such as, for example, whether or not to display invalidated options (and subsequently allow single selections overrides), whether or not user interface pages are presented as sequential pages or as grouped elements, whether or not pages/groups for each domain are to be displayed and where they are to be displayed, whether or not the domain is hidden (not included in the user interface), whether or not the domain is displayed as a read only form element (whereby the domain selection may only be made through configuration logic), whether or not to include local or specific domain and domain options representations, whether or not and which corresponding images are to be displayed with each option selection, whether or not and in which form an image element is to be updated with the corresponding image, whether or not and for which corresponding help pages are available for each domain and option selection, whether or not and in what form a status text to offer when navigation to each domain is included, and whether or not the domain is required for final product selection (order inclusion; This disclosure of Galea clearly teaches that data model contained in the GUI database 302 contains a plurality of entity types, that is, the database contains compiled product or server components (also referred to as domains and each of these domains contains descriptive binding domain elements to their respective GUI representations.*

Therefore, domains are entities and each entity describes a type of data structure such as various graphical user elements such as radio buttons, click boxes, input text fields etc.) **"wherein when an entity satisfies the set of conditions"** (Galea in view of Everett as applied to claim 20 above) **"the classification comprises classifying the entity as a particular entity type"** (Galea in view of Everett as applied to claim 20 above) **"for which a user interface element is created within a particular window of a set of windows in the description of the user interface, wherein the user interface element provides access to said entity through the user interface"** (Galea in view of Everett as applied to claim 20 above) and

"generating the user interface with at least one user interface element using the description of the user interface, wherein said user interface allows a user to transact with said database" (Galea, Figure 9; Note that user interface elements are created within a set of windows and users can access to the entities contained in the GUI database 302 via said user interface elements such as list box, multi-select list box, radio buttons, click boxes, and input text field ; Also note Cason, Column 10 Lines 51-67 in view of Galea, Column 5 Lines 51-59, i.e., *A user accesses a server through client 106, 108 via wide area network 112 in order to access items for sale (domains) in the product and GUI database. Each product or service configuration domain, together with the user interface tags, is compiled into a secure, binary compressed file format. After a user accesses the domain through client 106, 108, the compiled configuration model is downloaded to the browser of client 106, 108. The user*

interface is dynamically generated based upon the compiled domain tags; Also see **Carson** Column 11 Lines 1-9).

Claim 36 is rejected on the same basis as claim 27.

As per claim 37, Galea in view of Everett and further in view of Guyon and further in view of Cason teaches the limitation:

“before the receiving: sending preferences for the user interface, the preferences being utilized to create a customer user interface description based on the specified preferences” (Galea, Column 7 Lines 42-47, i.e., *A user accesses server 102 via WAN 112 from client 106. When the user makes a selection as to a domain (for example, an online camera store), the compiled interactive decision map 310 for the requested domain is downloaded to client 106. In addition, a client applet is downloaded to client 106. The client applet downloads the interactive decision map 310 for the particular domain and builds a multi-page graphical user interface (GUI) map by looping through the interactive decision map 310 and collecting array elements for each configuration domain*; Galea, Column 11 Lines 65-66, i.e., *Window 900 is displayed after a user selects a particular domain for viewing*; and Galea, Column 12 Lines 4-6, i.e., *After the user accesses a particular domain, the information is downloaded to the client from the server and the client applet downloaded from the server is used to create the display*;). The domain received from the client by the application server can be interpreted as the preference selected by the user and it is clear that the selected

domain will be utilized in creating a description of the user interface as shown in Figure 15. Also note Carson, that is, Cason, Column 10 Lines 51-67, i.e., *Change record key 226 processes a request by the user to edit and change a primary key value. This is a touchy point in data bases, **because everything is driven by the primary key(s).** Usually, controls are in place to prevent changing such primary keys because there are many pointers to that key. In accordance with this preferred embodiment of the invention, change record key 226 initiates a global change, and all pointers to the record identified by the primary key being changed are accessed and updated. For example, if a primary key is a name, and person identified by that name marries, then the key would need to be changed. In those instances where records are stored in user's database 218 in alphabetical, or some other such order, get next record key 230 and get previous record key 232 are provided for accessing the next and previous records, respectively; Cason Column 11 Lines 1-9, i.e., In response to the identification of the entity selected, meta data file 216 is accessed to determine which relationships exist for it. There will be one control established at terminal 219 for each relationship defined in meta data file 216 for the selected entity. **Just as each entity has one or more primary keys, relationships have keys that link entities. A control may be, for example, a push button, popup menu, icon, function key, number/enter key--whatever it takes to start an action;** Carson Column 12 Lines 5-25, i.e., *The user interface model describes the look and feel, which includes the type of control to be invoked, the type of window to be displayed, and the default values to be placed in the fields as the user defines a database model for some aspect of a**

business or other organization or process (such as the database example set forth in connection with FIG. 7) and thereafter works with that database **to add, delete, change records or modify the database structure itself**; Note that any changes to database entities and their respective keys as well as relationship(s) between database entities (including respective "types of controls") are stored in the "user Interface mode" of Cason, from which a graphical user interface is generated which allows the user to add, delete, change records or modify the database structure). Note Figure 8 of Cason wherein the user terminal is separated from the database. Also note Column 8 Lines 19-40, wherein "presentation manager" and "presentation manager code" on the client generates "graphical user interface" which represent the database.

As per claim 46, Galea in view of Everett and further in view of Guyon and further in view of Cason is directed to a system and teaches the limitations:

"a first computer system comprising a database having a data model and comprising a plurality of entities" (Galea, Figure 3 and Column 7 Lines 23-41, where it is clear that a data model or GUI database 302 is received by the modeler 304); and

"a server communicatively coupled to the database" (Galea, Column 5 Lines 11-15, i.e., *Figure 1 is an illustration of one embodiment for a distributed client-side e-commerce service system 100. Referring to Figure 1, server 100 is coupled to mass storage 104. Server 102 and mass storage device 104 are coupled via wide area network (WAN); and Galea, Figure 3 : 104 (which is mass storage device) which comprises product & GUI data 302) "for creating a description of a user interface*

that interacts with the database” (Galea, Column 5 Lines 51-59, i.e., *A user accesses a server through client 106, 108 via wide area network 112 in order to access items for sale (domains) in the product and GUI database. Each product or service configuration domain, together with the user interface tags, is compiled into a secure, binary compressed file format. After a user accesses the domain through client 106, 108, the compiled configuration model is downloaded to the browser of client 106, 108. The user interface is dynamically generated based upon the compiled domain tags*), **“the description being based upon a set of conditions that classify the the plurality of entities into entity types”** (Galea in view of Everett as applied to claim 20 above; Galea Column 5 Lines 23-50 in view of Cason Column 4 Lines 29-57 as applied to claim 20 above; Galea, Column 5 Lines 23-50, i.e., *In one embodiment, the databases contain compiled product or server components (domains). Each domain contains descriptive information which bind the domain elements to their respective GUI representations. Prerequisite tags may indicate the GUI element type such as, for example, single select list box, multi-select list box, radio buttons, click boxes, and input text field, or images that present the selections options for that domain. Optional domain tags determine the run time behavior of each GUI element such as, for example, whether. or not to display invalidated options (and subsequently allow single selections overrides), whether or not user interface pages are presented as sequential pages or as grouped elements, whether or not pages/groups for each domain are to be displayed and where they are to be displayed, whether or not the domain is hidden (not included in the user interface), whether or not the domain is displayed as a read only*

form element (whereby the domain selection may only be made through configuration logic), whether or not to include local or specific domain and domain options representations, whether or not and which corresponding images are to be displayed with each option selection, whether or not and in which form an image element is to be updated with the corresponding image, whether or not and for which corresponding help pages are available for each domain and option selection, whether or not and in what form a status text to offer when navigation to each domain is included, and whether or not the domain is required for final product selection (order inclusion; This disclosure of Galea clearly teaches that data model contained in the GUI database 302 contains a plurality of entity types, that is, the database contains compiled product or server components (also referred to as domains and each of these domains contains descriptive binding domain elements to their respective GUI representations. Therefore, domains are entities and each entity describes a type of data structure such as various graphical user elements such as radio buttons, click boxes, input text fields etc.)

"wherein the classification comprises classifying the entity as a particular entity type for which a user interface is created within a particular window of a set of windows in the description of the user interface" (Galea in view of Everett as applied to claim 20 above) **"wherein the user interface element provides access to said entity through the user interface"** (Galea in view of Everett; Galea Figure 9 as cited in claim 20 above; Galea Figure 8; entities (i.e., user elements) are determined and classified as *"Images"*, *"Navigation Frame" entities*, *"Dynamic Content Frame elements"*; Galea, Column 5 Lines 23-50, i.e., *Prerequisite tags may indicate the GUI*

element type such as, for example, single select **list box**, **multi-select list box**, **radio buttons**, **click boxes**, and **input text field**, or images that present the selections options for that domain).

As per claim 47, Galea in view of Everett and further in view of Guyon and further in view of Cason teaches the limitation:

“wherein the server obtains a current model of the database, the current data model reflecting any changes to the database up to when the current model is obtained, a current description of the user interface being created using the current data model of the database” (Galea, Column 6 Lines 20-34, i.e., *The distributed client-side e-commerce service system 100 allows for the **automatic generation** of a user interface for a particular product and/or service model “on-the-fly” such that a web user is always presented with the **latest** product or service selection choices*). In Column 10 Lines 44-57, Galea teaches database modeling. From these teachings, it can be inferred that Galea obtains a current data model of the database.

As Claim 48, Galea in view of Everett and further in view of Guyon and further in view of Cason teaches the limitation:

“wherein the server is in persistent communication with the database”
(Galea, Figure 3).

As per claim 49, Galea in view of Everett and further in view of Guyon and further in view of Cason teaches the limitation:

“wherein the server of the first computer system is communicatively coupled to the client of the second computer system via a network and distributes the created description to the client for enabling the client to generate the user interface” (Galea Figure 1) and **“distributes the created description to the first client for enabling the first client to generate the user interface”** (Galea, Column 5 Lines 51-59 and Column 6 Lines 20-34).

As per claim 50, Galea in view of Everett and further in view of Guyon and further in view of Cason teaches the limitation:

“wherein the server of the first computer system provides the first client an only point of access to the database of the first computer system” (Galea, Figure 1 and Figure 3).

As per claim 52, Galea in view of Everett and further in view of Guyon and further in view of Cason teaches the limitation teaches is directed to a computer and teaches the limitations:

a) **“a description of a data store specifying a user interface for transacting with data sets of the data store”** (Galea, Galea, Column 5 Lines 51-59, i.e., *A user accesses a server through client 106, 108 via wide area network 112 in order to access items for sale (domains) in the product and GUI database. Each product or service*

configuration domain, together with the user interface tags, is compiled into a secure, binary compressed file format. After a user accesses the domain through client 106, 108, the compiled configuration model is downloaded to the browser of client 106, 108. The user interface is dynamically generated based upon the compiled domain tags)

b) **“a browser for communicatively coupled to a server application of a different computer that provides a single point of access to the data store”**
(Galea, Figure 6, i.e., *Browser Application*); and

c) **“an application for generating user interface elements of the user interface by using said description and by retrieving a data set from said data store to populate at least one user interface element, said user interface elements for displaying in said browser”** (Galea, Galea, Column 5 Lines 51-59 i.e., *A user accesses a server through client 106, 108 via wide area network 112 in order to access items for sale (domains) in the product and GUI database. Each product or service configuration domain, together with the user interface tags, is compiled into a secure, binary compressed file format. After a user accesses the domain through client 106, 108, the compiled configuration model is downloaded to the browser of client 106, 108. The user interface is dynamically generated based upon the compiled domain tags; Galea Column 11 line 64 through Column 12 line 16, i.e., FIG. 9 illustrates an exemplary information block input window 900. Window 900 is displayed after a user selects a particular domain for viewing. The information used to draw or paint to the HTML page are shown in FIG. 9 are downloaded from server to client after a user accesses a particular domain. In the example of FIG. 9, the information is used to*

*display a medicine or drugstore e-commerce application. After the user accesses a particular domain, the information is downloaded to the client from the server and the client applet downloaded from the server is used to create the display. Window 900 includes a number of fields for selecting specific information contained in the downloaded interactive decision map file. FIG. 10 illustrates an exemplary interactive decision map window 1000. Decision map window 1000 is displayed after the user selects specific information from information block input window 800. Window 1000 includes a number of input and drop selection boxes and for choosing and selecting information downloaded from server 102) and "at least one user interface element for receiving queries for the data store" (Galea, Figure 11, i.e., SEARCH 1126 and Galea, Column 12 Lines 16-27, i.e., FIG. 11 illustrates an exemplary feature selection window 1100. Window 1100 includes a feature selection pop up window 1102 containing a number of feature selection pull down boxes 1104-1124. In addition, window 1100 also includes a **search box 1126** and a browse box 1128. After a domain file has been downloaded from the server to the client, the user may make selections of the various products from any of the pull down boxes (1104-1124). For example, the user may pull down picture quality box 1104 to give the selections shown at 1144. In this example, picture qualities may be selected based on poor, decent, good, very good, excellent or any combination deemed suitable).*

As per claim 53, Galea in view of Everett and further in view of Guyon and further in view of Cason teaches the limitation teaches the limitation:

“comprising a storage for storing the description, the browser and the application” (Galea Figures 1, 3, and 6).

As per claim 56, Galea in view of Everett and further in view of Guyon and further in view of Cason teaches the limitation:

“wherein said browser is a web browser” (Galea Figure 6), **“wherein said application is a distributed application running on said web browser”** (Galea, Figure 3).

As per claim 57, Official Note is taken that distributing an application, as an applet is notorious well known in the art, as can be seen Java applets on millions of web sites. Galea also teaches a Java runtime applet on client 106 in column 6 lines 60-62.

As per claim 58, Galea in view of Everett and further in view of Guyon and further in view of Cason teaches the limitation:

a) **“receiving a first request for a first user interface of a first client to transact with a first data store”** (Galea Column 5 Lines 51-55, i.e., *A user access a server through client 106, 108 via wide area network 112 in order to access items for sale (domains) in the product and GUI database. Each product or service configuration domain, together with the user interfaces tags , is compiled into a secure, binary compressed file format. And Galea, Column 5 Lines 10-22, i.e., FIG. 1 is an illustration of one embodiment for a distributed client-side e-commerce service system 100.*

Referring to FIG. 1, server 102 is coupled to mass storage device 104. Server 102 and mass storage device 104 are coupled via wide area network (WAN) 112 to a variety of clients 106 and 108. Wide area network 112 may be coupled to any of a variety of clients 106 and 108. In one embodiment, mass storage device 104 contains a product component and graphical user interface (GUI) database and an interactive decision map database. In an alternate embodiment, each of the databases may be contained in a separate mass storage devices 104);

b) **"generating a first description to supply to the first client in order for the first client to generate the first user interface"** (Galea, Column 4 Lines 1-5, i.e., *the constraint-based configuration file is mapped into a **plurality of pages** at a client, and a page of the **plurality of page** is updated directly from the constraint-based configuration file*; Column 5 Lines 19-22, i.e., *In an alternate embodiment, **each of the databases may be contained in a separate mass storage devices 104***; Column 5 Lines 23-26, i.e. *In one embodiment, the databases contain compiled product or service components (domains). Each domain contains descriptive information which bind the domain elements to their respective GUI representations*; Figure 12 and Figure 13; Column 12 Lines 28-50, which explains Figure 12 and Figure 13; Also note Figure 1 wherein *Client 106 (first client) and Client 108 (second client) can access the server for different user interfaces*));

c) **"receiving a second request for a second user interface to transact with a second data store"** (Galea Column 5 Lines 51-55, i.e., *A user access a server through client 106, 108 via wide area network 112 in order to access items for sale (domains) in*

the product and GUI database. Each product or service configuration domain, together with the user interfaces tags , is compiled into a secure, binary compressed file format. And Galea, Column 5 Lines 10-22, i.e., *FIG. 1 is an illustration of one embodiment for a distributed client-side e-commerce service system 100. Referring to FIG. 1, server 102 is coupled to mass storage device 104. Server 102 and mass storage device 104 are coupled via wide area network (WAN) 112 to a variety of clients 106 and 108. Wide area network 112 may be coupled to any of a variety of clients 106 and 108. In one embodiment, mass storage device 104 contains a product component and graphical user interface (GUI) database and an interactive decision map database. In an alternate embodiment, each of the databases may be contained in a separate mass storage devices 104;* Figure 12 and Figure 13; Column 12 Lines 28-50, which explains Figure 12 and Figure 13; Also note Figure 1 wherein *Client 106* (first client) and *Client 108* (second client) can access the server for different user interfaces); Particular note Galea's disclosure which states that ***"In an alternative embodiment, each of the databases may be contained in a separate mass storage devices 104,*** which clearly teaches that there are more than one data stores which different clients (users) could access;

and d) **"generating a second description to supply to the second client in order for the second client to generate the second user interface, wherein said first and second descriptions differ"** (Galea, Column 4 Lines 1-5, i.e., *the constraint-based configuration file is mapped into a plurality of pages at a client, and a page of the plurality of page is updated directly from the constraint-based configuration file;*

Column 5 Lines 19-22, i.e., *In an alternate embodiment, each of the databases may be contained in a separate mass storage devices 104*; Column 5 Lines 23-26, i.e. *In one embodiment, the databases contain compiled product or service components (domains). Each domain contains descriptive information which bind the domain elements to their respective GUI representations*; Figure 12 and Figure 13; Column 12 Lines 28-50, which explains Figure 12 and Figure 13; Also note Figure 1 wherein *Client 106* (first client) and *Client 108* (second client) can access the server for different user interfaces).

As per claim 59, Galea in view of Everett and further in view of Guyon and further in view of Cason teaches the limitation:

a) **"receiving a first request from a first user"** (Figure 1 wherein by way of *Client 106* (first client) and *Client 108* (second client), different users can access the server for different user interfaces) **"said user request comprising a set of preferences by the first user"** (Galea, Column 7 Lines 42-47, i.e., *A user accesses server 102 via WAN 112 from client 106. When the user makes a selection as to a domain (for example, an online camera store), the compiled interactive decision map 310 for the requested domain is downloaded to client 106. In addition, a client applet is downloaded to client 106. The client applet downloads the interactive decision map 310 for the particular domain and builds a multi-page graphical user interface (GUI) map by looping through the interactive decision map 310 and collecting array elements for each configuration domain*; Galea, Column 11 Lines 65-66, i.e., *Window 900 is displayed*

*after a user selects a particular domain for viewing; and Galea, Column 12 Lines 4-6, i.e., After the user accesses a particular domain, the information is downloaded to the client from the server and the client applet downloaded from the server is used to create the display;). The domain received from the client by the application server can be interpreted as the preference selected by the user and it is clear that the selected domain will be utilized in creating a description of the user interface as shown in Figure 15. Also note Carson, that is, Cason, Column 10 Lines 51-67, i.e., Change record key 226 processes a request by the user to edit and change a primary key value. This is a touchy point in data bases, **because everything is driven by the primary key(s).** Usually, controls are in place to prevent changing such primary keys because there are many pointers to that key. In accordance with this preferred embodiment of the invention, change record key 226 initiates a global change, and all pointers to the record identified by the primary key being changed are accessed and updated. For example, if a primary key is a name, and person identified by that name marries, then the key would need to be changed. In those instances where records are stored in user's database 218 in alphabetical, or some other such order, get next record key 230 and get previous record key 232 are provided for accessing the next and previous records, respectively; **Cason** Column 11 Lines 1-9, i.e., In response to the identification of the entity selected, meta data file 216 is accessed to determine which relationships exist for it. There will be one control established at terminal 219 for each relationship defined in meta data file 216 for the selected entity. **Just as each entity has one or more primary keys, relationships have keys that link entities. A control may be, for***

example, a push button, popup menu, icon, function key, number/enter key-- whatever it takes to start an action; Carson Column 12 Lines 5-25, i.e., **The user interface model describes the look and feel, which includes the type of control to be invoked, the type of window to be displayed, and the default values to be placed in the fields as the user defines a database model for some aspect of a business or other organization or process** (such as the database example set forth in connection with FIG. 7) and thereafter works with that database **to add, delete, change records or modify the database structure itself;** Note that any changes to database entities and their respective keys as well as relationship(s) between database entities (including respective "types of controls") are stored in the "user Interface mode" of Cason, from which a graphical user interface is generated which allows the user to add, delete, change records or modify the database structure). Note Figure 8 of Cason wherein the user terminal is separated from the database. Also note Column 8 Lines 19-40, wherein "presentation manager" and "presentation manager code" on the client generates "graphical user interface" which represent the database);

b) **"receiving a second request from a second user"** (Figure 1 wherein by way of *Client 106* (first client) and *Client 108* (second client), different users can access the server for different user interfaces) **"said second request comprising a set of preferences specified by the second user, the first and second users specifying different sets of preferences"** (Figure 1 wherein by way of *Client 106* (first client) and *Client 108* (second client), different users can access the server for different user interfaces) ; Galea, Column 7 Lines 42-47, i.e., **A user accesses server 102 via WAN**

112 from client 106. When the user makes a selection as to a domain (for example, an online camera store), the compiled interactive decision map 310 for the requested domain is downloaded to client 106. In addition, a client applet is downloaded to client 106. The client applet downloads the interactive decision map 310 for the particular domain and builds a multi-page graphical user interface (GUI) map by looping through the interactive decision map 310 and collecting array elements for each configuration domain; Galea, Column 11 Lines 65-66, i.e., Window 900 is displayed after a user selects a particular domain for viewing; and Galea, Column 12 Lines 4-6, i.e., After the user accesses a particular domain, the information is downloaded to the client from the server and the client applet downloaded from the server is used to create the display;).

The domain received from the client by the application server can be interpreted as the preference selected by the user and it is clear that the selected domain will be utilized in creating a description of the user interface as shown in Figure 15. Also note Carson, that is, Cason, Column 10 Lines 51-67, i.e., Change record key 226 processes a request by the user to edit and change a primary key value. This is a touchy point in data bases, **because everything is driven by the primary key(s)**. Usually, controls are in place to prevent changing such primary keys because there are many pointers to that key. In accordance with this preferred embodiment of the invention, change record key 226 initiates a global change, and all pointers to the record identified by the primary key being changed are accessed and updated. For example, if a primary key is a name, and person identified by that name marries, then the key would need to be changed. In those instances where records are stored in user's database 218 in alphabetical, or

some other such order, get next record key 230 and get previous record key 232 are provided for accessing the next and previous records, respectively; **Cason** Column 11 Lines 1-9, i.e., *In response to the identification of the entity selected, meta data file 216 is accessed to determine which relationships exist for it. There will be one control established at terminal 219 for each relationship defined in meta data file 216 for the selected entity. Just as each entity has one or more primary keys, relationships have keys that link entities. A control may be, for example, a push button, popup menu, icon, function key, number/enter key--whatever it takes to start an action;* **Carson** Column 12 Lines 5-25, i.e., *The user interface model describes the look and feel, which includes the type of control to be invoked, the type of window to be displayed, and the default values to be placed in the fields as the user defines a database model for some aspect of a business or other organization or process (such as the database example set forth in connection with FIG. 7) and thereafter works with that database to add, delete, change records or modify the database structure itself;* Note that any changes to database entities and their respective keys as well as relationship(s) between database entities (including respective "types of controls") are stored in the "user Interface mode" of Cason, from which a graphical user interface is generated which allows the user to add, delete, change records or modify the database structure). Note Figure 8 of Cason wherein the user terminal is separated from the database. Also note Column 8 Lines 19-40, wherein "presentation manager" and "presentation manager code" on the client generates "graphical user interface" which represent the database);

c) **“generating (i) a first description customized according to the first user’s preferences to supply to the first user and (ii) a second description customized to the second user’s preferences to supply to the second user”** (Galea, Figure 1 wherein by way of *Client 106* (first client) and *Client 108* (second client), different users can access the server for different user interfaces; Galea, Column 4 Lines 1-5, i.e., *the constraint-based configuration file is mapped into a plurality of pages at a client, and a page of the plurality of page is updated directly from the constraint-based configuration file*; Column 5 Lines 19-22, i.e., *In an alternate embodiment, each of the databases may be contained in a separate mass storage devices 104*; Column 5 Lines 23-26, i.e. *In one embodiment, the databases contain compiled product or service components (domains). Each domain contains descriptive information which bind the domain elements to their respective GUI representations*; Figure 12 and Figure 13; Column 12 Lines 28-50, which explains Figure 12 and Figure 13; Also note Figure 1 wherein *Client 106* (first client) and *Client 108* (second client) can access the server for different user interfaces, wherein the first and second descriptions are different based on the roles of the users) **“wherein the first and second descriptions are different based on the set of preferences specified by the users”** (Galea and Cason as cited above).

As per claim 60, Galea in view of Everett and further in view of Guyon and further in view of Cason teaches the limitation:

“wherein the first and second user interfaces comprises at least two interface elements for facilitating data transactions” (see Galea, Figure 12 and 13,

wherein more than two interface elements are displayed), "wherein said first interface comprises at least one or more user interface elements than said second user interface" (Galea Figure 1 describes 2 clients and Galea Figures 12 and 13 describes a plurality of interface elements on each generated user interfaces. Depending on domains, numbers of user interface elements are different. As such, one interface always has more interface elements than other interface).

As per claim 61, Galea in view of Everett and further in view of Guyon and further in view of Cason teaches the limitation:

"wherein first and second user interfaces are displayed in an application running different computers" (Galea, Figure 1, *Client 106* (first client) and *Client 108* (second client)).

As per claim 62, Galea in view of Everett and further in view of Guyon and further in view of Cason teaches the limitation:

"wherein the application is web browser" (Galea, Galea, Column 5 Lines 51-59 i.e., *A user accesses a server through client 106, 108 via wide area network 112 in order to access items for sale (domains) in the product and GUI database. Each product or service configuration domain, together with the user interface tags, is compiled into a secure, binary compressed file format. After a user accesses the domain through client 106, 108, the compiled configuration model is downloaded to the browser of client 106, 108.*).

As per claim 63 Galea in view of Everett and further in view of Guyon and further in view of Cason teaches the limitation:

"wherein the first and second data stores are the different data stores, wherein the first request is received from a first user while the second request is received from a second user different than the first, wherein the description is supplied to the first user while the second description is supplied to the second user" (Galea, Column 4 Lines 1-5, i.e., *the constraint-based configuration file is mapped into a **plurality of pages** at a client, and a page of the **plurality of page** is updated directly from the constraint-based configuration file*; Column 5 Lines 19-22, i.e., *In an alternate embodiment, **each of the databases may be contained in a separate mass storage devices 104***; Column 5 Lines 23-26, i.e. *In one embodiment, the databases contain compiled product or service components (domains). Each domain contains descriptive information which bind the domain elements to their respective GUI representations*; Figure 12 and Figure 13; Column 12 Lines 28-50, which explains Figure 12 and Figure 13; Also note Figure 1 wherein *Client 106* (first client) and *Client 108* (second client) can access the server for different user interfaces).

As per claim 75, Galea in view of Everett and further in view of Guyon and further in view of Cason teaches the limitation:

"wherein said request from said client initiates the classification of the entities into entity types" (Cason, Figure 10A i.e., **INITIALIZATIONS & DEFAULTS**

and Column 9 Lines 22-26, i.e., *Referring now to FIG. 10, a block diagram illustration of a preferred embodiment of the invention is described from the view of the user at terminal 200. In this embodiment, **initialization and defaults block 202***).

As per claim 76, Galea in view of Everett and further in view of Guyon and further in view of Cason teaches the limitation:

"wherein each entity corresponds to at least one table of said database"

(Cason, Figure 1, i.e., 104 (which is a table) and 102 (which is a table)).

As per claim 77, Galea in view of Everett and further in view of Guyon and further in view of Cason teaches the limitation:

"further comprising a set of instructions for sending said description to said client in order to allow said client to generate elements of said user interface from said description" (Galea, Column 5 Lines 51-59, i.e., *A user accesses a server through client 106, 108 via wide area network 112 in order to access items for sale (domains) in the product and GUI database*; Cason, Column 10 Lines 51-67; Cason Column 11 lines 1-9), "said user interface elements allowing a user to transact with the database" (Cason Column 11 lines 1-9, i.e., *"thereafter works with that database to add, delete, change records or modify the database structure itself"*).

As per claim 78, Galea in view of Everett and further in view of Guyon and further in view of Cason teaches the limitation:

“wherein an entity describes a data object of said database through properties comprising at least one of an attribute and a relationship” (Cason, Cason Column 4 Lines 29-57, i.e., *In an ERwin diagram, the entity is represented by drawing a box 280 with the name of the entity on top and all of the attributes of the entity inside the box. The horizontal line 281 in the entity box divides the attributes into two sets: keys and non-keys (or, data). The area above the line is the key area 282, and the area below the line is the data area 284. The key is the set of attributes that identifies an entity; Cason Column 11 Lines 1-9, i.e., In response to the identification of the entity selected, meta data file 216 is accessed to determine which relationships exist for it. There will be one control established at terminal 219 for each relationship defined in meta data file 216 for the selected entity. Just as each entity has one or more primary keys, relationships have keys that link entities).*

As per claim 79, Galea in view of Everett and further in view of Guyon and further in view of Cason teaches the limitation:

“wherein the entity corresponds to a table of said database” (Cason, Figure 1, i.e., 104 (which is a table) and 102 (which is a table)).

As per claim 80, Galea in view of Everett and further in view of Guyon and further in view of Cason teaches the limitation:

“from a client, receiving a request for the user interface” (Galea, Column 5, Lines 51-53, i.e., *a user accesses a server through client 106, 108 via wide area*

*network 112 in order to access items for sale (domains) in the product and GUI database; Cason, Figure 1 and Column 3 Lines 29-31, i.e., Referring further to Fig. 1, the specific panel displays 122 and 124 provide examples of **graphical user interface(GUI) representations of the data model entities** 102 and 104, respectively; Cason, Figure 10A i.e., **INITIALIZATIONS & DEFAULTS**);*

“in response to said request, obtaining a current data model describing a plurality of tales of the database” (Galea, Figure 3 and Column7 Lines 23-41, where it is clear that a data model or GUI database 302 is received by the modeler 304; Galea, Column 5 Lines 51-59, i.e., A user accesses a server through client 106, 108 via wide area network 112 in order to access items for sale (domains) in the product and GUI database. Each product or service configuration domain, together with the user interface tags, is compiled into a secure, binary compressed file format. After a user accesses the domain through client 106, 108, the compiled configuration model is downloaded to the browser of client 106, 108. The user interface is dynamically generated based upon the compiled domain tags; Cason, Figure 1 and Column 3 Lines 29-31, i.e., Referring further to Fig. 1, the specific panel displays 122 and 124 provide examples of graphical user interface(GUI) representations of the data model entities 102 and 104, respectively; Cason, Figure 10A i.e., **INITIALIZATIONS & DEFAULTS** and Column 9 Lines 22-26, i.e., Referring now to FIG. 10, a block diagram illustration of a preferred embodiment of the invention is described from the view of the user at terminal 200. In this embodiment, initialization and defaults block 202 references online help file 204, screen layouts & bitmaps 206, user custom table 217, and start from

selection list 210; Note that by initializing the system/method, the user herein is requesting to obtain a graphical interface representation of database models 122 and 124);

"automatically generating a description of said user interface based on the current data model" (Galea, Column 5 Lines 51-59, i.e., *A user accesses a server through client 106, 108 via wide area network 112 in order to access items for sale (domains) in the product and GUI database. Each product or service configuration domain, together with the user interface tags, is compiled into a secure, binary compressed file format. After a user accesses the domain through client 106, 108, the compiled configuration model is downloaded to the browser of client 106, 108. The user interface is dynamically generated based upon the compiled domain tags;* Cason, Column 11 Lines 10-31, i.e., *In accordance with the invention, each user's database 218 will include one or more description entities (see, for example, table definition 304 and field definition 306, FIG. 11) and each relationship is given a type from a set including the type "description", as will be more fully described in connection with FIGS. 11-16. These entities include a description of a particular record containing information in a plurality of leafs, one leaf for each record, necessary to enforce the cardinality rules. In this embodiment, invoke a relationship control 236 checks the type of the relationship linking the entity selected to the entity displayed and, as shown at test 240, invokes page editor 238 if the relationship has type "text description". Page input editor 238 opens at terminal 219, allowing the user to enter to the description entity instructions which define the cardinality rules. Page input editor*

238 uniquely numbers the lines of the leaf being edited thus providing keys to the lines of data in the description, closes out the window and returns control to record edit interface 220. Alternatively, the user may activate cancel 250 key 250 to return control to invoke relationships 236 without saving any changes made to the selected leaf of the description entity being edited; **Carson** Column 12 Lines 5-25, i.e., Referring now to FIGS. 11-16, the ERwin diagrams describing the preferred embodiment of the invention will be described. These FIGS. 11-16 include entity-attribute-relationship (EAR) models of FIGS. 11, 13 and 15 and physical models of FIGS. 12, 14 and 16. FIGS. 11 and 12 describe the database model. FIGS. 13 and 14 describe **the user interface model**. FIGS. 15 and 16 combine the database and user interface models. The EAR model contains the database model which describes the invention. A physical model describes an actual implementation of an EAR model of the invention for a given database, and includes table names, field names within tables, and their type and size. The relationships in a physical model are the same as in the EAR model. **The user interface model describes the look and feel, which includes the "type of control" to be invoked, the type of window to be displayed, and the default values to be placed in the fields as the user defines a database model for some aspect of a business or other organization or process** (such as the database example set forth in connection with FIG. 7) and thereafter works with that database to add, delete, change records or modify the database structure itself; Note herein that "the user interface model" of Carson maps to the "description of the user interface" of the claimed invention. Said "user interface model" of Carson is derived based on "keys" which

uniquely identifies database entities from a single record to a table. As such, said "user interface model" includes "descriptions" (Carson Column 11 Lines 10-31) and are derived based on keys);

"sending said description to said client" (Galea, Column 5 Lines 51-59, i.e. *After a user accesses the domain through client 106, 108, the compiled configuration model is downloaded to the browser of client 106, 108*);

"from said client, receiving a request for data from at least one table of the database to populate at least one user interface elements of said user interface" (Galea Column 11 line 64 through Column 12 line 16, i.e., FIG. 9 illustrates an exemplary information block input window 900. Window 900 is displayed after a user selects a particular domain for viewing. The information used to draw or paint to the HTML page are shown in FIG. 9 are downloaded from server to client after a user accesses a particular domain. In the example of FIG. 9, the information is used to display a medicine or drugstore e-commerce application. After the user accesses a particular domain, the information is downloaded to the client from the server and the client applet downloaded from the server is used to create the display. Window 900 includes a number of fields for selecting specific information contained in the downloaded interactive decision map file. FIG. 10 illustrates an exemplary interactive decision map window 1000. Decision map window 1000 is displayed after the user selects specific information from information block input window 800. Window 1000 includes a number of input and drop selection boxes and for choosing and selecting information downloaded from server 102);

"retrieving said data from the database" (*Galea* Column 11 line 64 through Column 12 line 16, i.e., FIG. 9 illustrates an exemplary information block input window 900. Window 900 is displayed after a user selects a particular domain for viewing. The information used to draw or paint to the HTML page are shown in FIG. 9 are downloaded from server to client after a user accesses a particular domain. In the example of FIG. 9, the information is used to display a medicine or drugstore e-commerce application. After the user accesses a particular domain, the information is downloaded to the client from the server and the client applet downloaded from the server is used to create the display. Window 900 includes a number of fields for selecting specific information contained in the downloaded interactive decision map file. FIG. 10 illustrates an exemplary interactive decision map window 1000. Decision map window 1000 is displayed after the user selects specific information from information block input window 800. Window 1000 includes a number of input and drop selection boxes and for choosing and selecting information downloaded from server 102) and

"sending said data to said client in order to allow said client to populate said user interface element, said client enabled thereupon to generate said user interfaced which permits a user of said client to transact with said database"

(*Galea* Column 11 line 64 through Column 12 line 16, i.e., FIG. 9 illustrates an exemplary information block input window 900. Window 900 is displayed after a user selects a particular domain for viewing. The information used to draw or paint to the HTML page are shown in FIG. 9 are downloaded from server to client after a user accesses a particular domain. In the example of FIG. 9, the information is used to

display a medicine or drugstore e-commerce application. After the user accesses a particular domain, the information is downloaded to the client from the server and the client applet downloaded from the server is used to create the display. Window 900 includes a number of fields for selecting specific information contained in the downloaded interactive decision map file. FIG. 10 illustrates an exemplary interactive decision map window 1000. Decision map window 1000 is displayed after the user selects specific information from information block input window 800. Window 1000 includes a number of input and drop selection boxes and for choosing and selecting information downloaded from server 102 ; Cason Column 11 lines 1-9, i.e., “thereafter works with that database to add, delete, change records or modify the database structure itself”).

As per claim 81, Galea in view of Everett and further in view of Guyon and further in view of Cason teaches the limitation:

“after sending said data to said client, receiving from the user of said data, a query for data stored in said database” (Galea, Figure 11, i.e., *SEARCH 1126* and Galea, Column 12 Lines 16-27, i.e., *FIG. 11 illustrates an exemplary feature selection window 1100. Window 1100 includes a feature selection pop up window 1102 containing a number of feature selection pull down boxes 1104-1124. In addition, window 1100 also includes a search box 1126 and a browse box 1128. After a domain file has been down loaded from the server to the client, the user may make selections of the various products from any of the pull down boxes (1104-1124). For example, the*

user may pull down picture quality box 1104 to give the selections shown at 1144. In this example, picture qualities may be selected based on poor, decent, good, very good, excellent or any combination deemed suitable).

As per claim 82, Galea in view of Everett and further in view of Guyon and further in view of Cason teaches the limitation:

“wherein said request for data to populate the user interface element is received automatically from the client prior to receiving a transaction or a query from the user of said client” (*Galea* Column 11 line 64 through Column 12 line 16, i.e., FIG. 9 illustrates an exemplary information block input window 900. Window 900 is displayed after a user selects a particular domain for viewing. The information used to draw or paint to the HTML page are shown in FIG. 9 are downloaded from server to client after a user accesses a particular domain. In the example of FIG. 9, the information is used to display a medicine or drugstore e-commerce application. After the user accesses a particular domain, the information is downloaded to the client from the server and the client applet downloaded from the server is used to create the display. Window 900 includes a number of fields for selecting specific information contained in the downloaded interactive decision map file. FIG. 10 illustrates an exemplary interactive decision map window 1000. Decision map window 1000 is displayed after the user selects specific information from information block input window 800. Window 1000 includes a number of input and drop selection boxes and for choosing and selecting information downloaded from server 102 ; Cason Column 11 lines 1-9, i.e.,

"thereafter works with that database to add, delete, change records or modify the database structure itself").

As per claim 83, Galea in view of Everett and further in view of Guyon and further in view of Cason teaches the limitation:

"wherein said creating said description comprises dynamically generating a description of said user interface based on an analysis of said data model in response to the request for the user interface from said client" (Galea, Column 6 Lines 20-34, i.e., *The distributed client-side e-commerce service system 100 allows for the automatic generation of a user interface for a particular product and/or service model "on-the-fly" such that a web user is always presented with the latest product or service selection choices*).

Claim 87 is rejected on the same basis as claim 59.

As per claim 88, Galea in view of Everett and further in view of Guyon and further in view of Cason teaches the limitation:

"before receiving the data stored in said data store, automatically sending a request for the data stored in said data store in order to populate the user interface element" (Galea Column 11 line 64 through Column 12 line 16, i.e., FIG. 9 illustrates an exemplary information block input window 900. Window 900 is displayed

after a user selects a particular domain for viewing. The information used to draw or paint to the HTML page are shown in FIG. 9 are downloaded from server to client after a user accesses a particular domain. In the example of FIG. 9, the information is used to display a medicine or drugstore e-commerce application. After the user accesses a particular domain, the information is downloaded to the client from the server and the client applet downloaded from the server is used to create the display. Window 900 includes a number of fields for selecting specific information contained in the downloaded interactive decision map file. FIG. 10 illustrates an exemplary interactive decision map window 1000. Decision map window 1000 is displayed after the user selects specific information from information block input window 800. Window 1000 includes a number of input and drop selection boxes and for choosing and selecting information downloaded from server).

9. Claims 21 and 22 are rejected under 35 U.S.C. 103(a) as being unpatentable over Galea in view of Everett and further in view of Guyon and further in view of Cason and further in view of in view of Roth (U.S. Patent Number 6564218).

As per claim 21, Cason in view of Galea teaches the limitation:

"each entity in the data model describes a type of data object associated with the database" (Galea, Column 5 Lines 23-50 in view of Cason Column 4 Lines 29-57 as applied to claim 20 above) and

"the classifying produces the particular entity type for a first group of data objects and a second different entity type of second a second group of data

objects, the data objects in the first group of data objects being updated in the database more frequently than the data objects in the second groups of data objects" (Galea in view of Everett as applied to claim 20 above and Cason Column 4 Lines 29-57).

However, Cason in view of Galea does not explicitly teach the limitation: **"the data objects in the first group of data objects being updated in the database more frequently than the data objects in the second groups of data objects"**.

On the other hand, Roth teaches the limitation:

"the data objects in the first group of data objects being updated in the database more frequently than the data objects in the second group of data objects" (Roth, Column 4 Lines 54-57, i.e., *Advantageously, said one or more supersets to be used for said validity check are selected from said specified supersets on the basis of a priori knowledge of supersets least likely to be updated*). Roth teaches supersets of data objects and some of said supersets are more frequently update while others are less frequently updated.

At the invention was made, it would have been obvious to a person of ordinary skill in the to add the feature of updating data objects/entities, of which some are more frequently updated and some are less frequently updated, as taught by Roth, to the method of Cason in view of Cason so that, in the resultant method, the classifying would produce the first entity type for a first group of data objects and a second entity type for a second group of data objects, the data objects in the first group of data objects being updated in the database more frequently than the data objects in the second group of

data objects. One would have been motivated to do so in order to improve overall speed of retrieval/updates of digital information (Roth, Column 4 Lines 39-43).

As per claim 22, Galea in view of Everett and further in view of Guyon and further in view of Cason and further in view of Roth teaches the limitations:

“wherein the particular entity type is a Main entity type and the second different entity type is an Enumeration type” (Roth, Column 8 Lines 65 through Column 9 Line 5, i.e., *As **changes typically affect some parameters with a higher probability than others**, the version comparison can be optimized by starting the comparison using the parameter with **the least probability of change**, e.g. **starting with the parameter “country”**, provided that data are changed on a country by country basis).*

10. Claim 23, 32, and 51 are rejected under 35 U.S.C. 103(a) as being unpatentable over Galea in view of Everett and further in view of Guyon and further in view of Cason and further in view of Sexton et al., (hereinafter “Sexton”, (U.S. Patent Number 7093263).

As per claim 23, Galea in view of Everett and further in view of Guyon and further in view of Cason does not explicitly teach the limitation: “wherein the description is a generic description configured to be interpreted in different platforms or operating environments”.

On the other hand, Sexton teaches the limitation:

"wherein the description is a generic description configured to be interpreted in different platforms or operating environments" (Sexton, Column 3 Lines 40-52, i.e., In addition, instructions can automatically be generated for getting and setting values in the object in the platform-independent format, thereby diminishing the reliance on manually coding the operations. As a result, errors are reduced and the code is more maintainable).

At the time the invention was made, it would have been obvious to a person of ordinary skill in the art to add the feature of employing generic descriptions/instructions which can be interpreted in different operating environments, as taught by Sexton, to the method of Galea in view of Everett and further in view of Guyon and further in view of Cason so that, in the resultant method, the description would be a generic description which is to be interpreted in different platforms. One would have been motivated to do so in order to make the code/instructions/descriptions portable, which is a well-known practice in the art, such as Java code.

Claim 32 is rejected on the same basis as claim 23.

As per claim 51, Galea in view of Everett and further in view of Guyon and further in view of Cason and further in view of Sexton teaches the limitations:

"wherein the server of the first computer system is communicatively coupled, via the network, to a client of a third computer system" (Galea Figure 1 in view of Sexton Figure 1) **"having a different platform or operating environment than**

the client of the second computer system” (Column 9 Lines 3-6, i.e., *For example, if a platform does not have a 4-byte primitive integer (e.g., on 64-bit machines such as Cray YP-1), and “distributes the created description to the client of the third computer system for enabling the client of the third computer system to generate the user interface”* (Galea, Column 5 Lines 51-59 and Column 6 Lines 20-34).

11. Claim 38 is rejected under 35 U.S.C. 103(a) as being unpatentable over Galea in view of Everett and further in view of Guyon and further in view of Cason and further view of Mashayekhi (U.S. Patent Number 5818936).

As per claim 38, Galea in view of Everett and further in view of Guyon and further in view of Cason does not explicitly teach the limitation: “before the receiving: sending authentication information”.

On the other hand, Mashayekhi teaches the limitation:

“before the receiving: sending authentication information” (Mashayekhi, Column 7 Lines 10-30, i.e., *when a user 201 attempts to access a particular application program, such as a local application 240 or network-based application program 236, the particular application program requires that the user be authenticated prior to accessing its processes or data).*

At the time the invention was made, it would have been obvious to a person of ordinary skill to add the feature of authenticating a user access, as taught by Mashayekhi, to the method of Galea in view of Cason so that the resultant method

would comprise sending authentication information. One would have been motivated to do so in order to provide security to the database, which is well known in the art.

Allowable Subject Matter

12. Claims 42-44 and 84-86 are allowed. The following is the statement of reasons for the indication of allowable subject matter.

As per claim 42, the prior art of record, alone or in combination, does not teach or fairly suggest the combination steps as recited in the claim. Galea in view of Everett and further in view of Guyon and further in view of Cason does not teach the following limitation(s):

"(ii) a second entity type for which a user interface element is created within a second window of the set of windows in the description of the user interface in order to access said entity upon determining that the entity does not satisfy the set of conditions".

The dependent claims of claim 42, being definite, further limiting, and fully enabled by the specification are also allowed.

As per claim 84, the prior art of record, alone or in combination, does not teach or fairly suggest the combination steps as recited in the claim. Galea in view of Everett and

further in view of Guyon and further in view of Cason does not teach the following limitation(s):

"in response to the request, dynamically generating a description of said user interface (i) by using a set of heuristics to select a set of data entities from the plurality of data entities for access by the client through a primary window of the description based on a relevancy of the set of data entities to the client, and (ii) by providing a set of task operable on each data entity".

The dependent claims of claim 84, being definite, further limiting, and fully enabled by the specification are also allowed.

Conclusion

13. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of

the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Contact Information

14. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Dennis Myint whose telephone number is (571) 272-5629. The examiner can normally be reached on 8:30AM-5:30PM Monday-Friday.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John Breene can be reached on (571) 272-4107. The fax phone number for the organization where this application or proceeding is assigned is 571-273-5629.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/dennis myint/

Dennis Myint
Examiner AU-2162

/Jean B. Fleurantin/
Primary Examiner, Art Unit 2162